LING/C SC/PSYC 438/538

Lecture 1

Prof. Sandiway Fong

Contents

Computation and Language

- We assume human language can be modeled as a formal system

  Research Question:

  - What is the nature of this formal system for human language?
  - How is it different from other (mathematical) formal systems?

- What we study here has much to do with formal systems

  Examples:

  - programming (languages)
  - regular expressions (regex), finite state automata (FSA)
  - context-free (CFG) and context-sensitive grammars
  - parse trees, word dependencies, sets

Computation and Language

- Why only us?
  - 50 billion species since life emerged on earth almost 4 billion years ago (Mayr, 1995)
  - Modern humans a recent arrival (300TYA) through a remarkable series of accidents
- Language and thought is a species-specific property (Chomsky)
  - closest relatives to modern humans have about the same auditory system
  - only a human infant reflexively develops complex systems for constructing and expressing thought
- Language shows peculiar properties
  - not arbitrary (there are rules: man-made or otherwise?)
  - **Research question**: why is it the way it is, and not some other way?

Syllabus

Description of Course

- An introductory level course at the advanced level for computational linguistics. Required core course for the Master's Human Language Technology (HLT) program.

Course Pre-requisites

- 438: LING 388 or familiarity with one or more of the following: formal languages, syntax, data structures, or compilers.
- 538: no formal pre-requisites.

Syllabus

Instructor and Contact Information

- Instructor: Sandiway Fong (sandiway@arizona.edu, submit homeworks here)
- Homepage: sandiway.arizona.edu
- Dept. of Linguistics Office: Douglass 311

Hours:

- make appointments by email or drop by my office

Meet:

- Rm 206, Psychology Building: 9:30-10:45am Tuesdays/Thursdays
- *another good time to ask questions is right after class!*

Syllabus

Course Format and Teaching Methods

- Lecture with slides. Panopto videos (when available) for lecture review.
- All homeworks will be introduced and reviewed in class.

Course Objectives

Topics covered include:

- Introductory programming relevant to computational linguistics in two or more programming languages. We will use Perl, Python and Prolog this semester.
- Introduction to a range of topics in computational linguistics, see detailed list of topics later below.

Syllabus

Course Learning Outcomes

After completing this course, students will:

1. Have acquired the ability to read and write programs in two or more programming languages.

- Relates to Linguistics Department HLT program outcome #1.

2. Be familiar with basic concepts, techniques and applications in computational linguistics.

- Relates to Linguistics Department HLT program outcome #2 and Linguistics Department Undergraduate program outcome #1.

3. **538-only**: be able to present and explain advanced concepts in computational linguistics. (See chapter presentation requirement.)

4. Be equipped to take more advanced classes in computational linguistics, e.g. 581 (Spring) or 439/539 (Statistical NLP).

Syllabus

Absence and Class Participation Policy

- I expect you to attend lectures (though attendance will not be taken).
- The UA's policy concerning Class Attendance, Participation, and Administrative Drops is available at: http://catalog.arizona.edu/policy/class-attendance-participation-and-administrative-drop.
- Tell me ahead of time so we can make alternative arrangements in the case of missed homeworks. **No homework will be accepted late. Explained below.**
- Absences pre-approved by the UA Dean of Students (or Dean Designee) will be honored. See: https://deanofstudents.arizona.edu/absences.
- The UA policy regarding absences for any sincerely held religious belief, observance or practice will be accommodated where reasonable, http://policy.arizona.edu/human-resources/religious-accommodation-policy.

Syllabus

Accessibility and Accommodations

- At the University of Arizona, we strive to make learning experiences as accessible as possible. If you anticipate or experience barriers based on disability or pregnancy, please contact the Disability Resource Center (520-621-3268, **https://drc.arizona.edu/)** to establish reasonable accommodations.

UA Nondiscrimination and Anti-harassment Policy

- The University is committed to creating and maintaining an environment free of discrimination; see http://policy.arizona.edu/human-resources/nondiscrimination-and-anti-harassment-policy.

Syllabus

Required Text

- 438: None.
- 538 Presentations: *Speech and Language Processing*, Jurafsky & Martin, draft 3rd edition (PDF available).

Required or Special Materials

- All necessary software will be available online at no cost to the student.
- However, students are expected to either have a laptop/desktop capable of handling homework and classwork, or make use of UA lab computers (?)
- Mac, PC (Windows 11) or Linux.

Syllabus

Assignments and Examinations: Schedule/Due Dates

- All homeworks will be introduced **and reviewed** in class.
- Homework submissions by email to me.
- Late homeworks will be not accepted since all homeworks will be solved in class.
- Quick homeworks are normally due at midnight before the next class, and are generally assigned in class on a **Tuesday** and due **Wednesday** midnight (before **Thursday** class).
- Homeworks not categorized as quick are normally assigned in class on a **Thursday** and due the following **Sunday** or **Monday** midnight (before next **Tuesday's** class). (Some longer homeworks may have an extended due date.)
- Students can expect a total of 10-14 homeworks over the course.

Syllabus

Final Examination or Project

- No examinations, e.g. mid-term or final, are scheduled for this course.

Grading Scale and Policies

- **438**:
  - 100% of the grade comes from the homework assignments.
- **538**:
  - 75% of the grade comes from the homework assignments (possibly a superset of the 438 assignments), 25% of the grade comes from a textbook chapter presentation.
- Requests for incomplete (I) or withdrawal (W) must be made in accordance with University policies, which are available at http://catalog.arizona.edu/policy/grades-and-grading-system#incomplete and http://catalog.arizona.edu/policy/grades-and-grading-system#Withdrawal respectively.

Syllabus

Scheduled Topics/Activities

- Topics will be drawn from the following:
- – Programming Languages: Perl and Python
- – Regular Expressions (Theoretical and practical)
- – Automata (Finite State) and Transducers (Finite State)
- – Programming Language: Prolog (definite clause grammars)
- – NLTK (Natural Language Toolkit)
- – Part of Speech (POS) Tagging
- – Stemming (Morphology)
- – Edit Distance (Spelling)
- – Grammars (Regular, Context-free)
- – Parsing (Syntax trees, dependency trees, algorithms)
- – *and more ...*

Syllabus

Code of Academic Integrity

- You may discuss homework questions with anyone (or anything).
- You may look things up on the web and use answers found therein; however, you must write it up yourself (in your own words/own code *etc*.).
- You must cite all (web) references including ChatGPT, and your classmates (in the case of shared discussion).
- Students are encouraged to share intellectual views and discuss freely the principles and applications of course materials.
- However, graded work/exercises must be the product of independent effort unless otherwise instructed.
- Students are expected to adhere to the UA Code of Academic Integrity as described in the UA General Catalog. See: http://deanofstudents.arizona.edu/academic-integrity/students/academic-integrity.

Syllabus

Subject to Change Statement

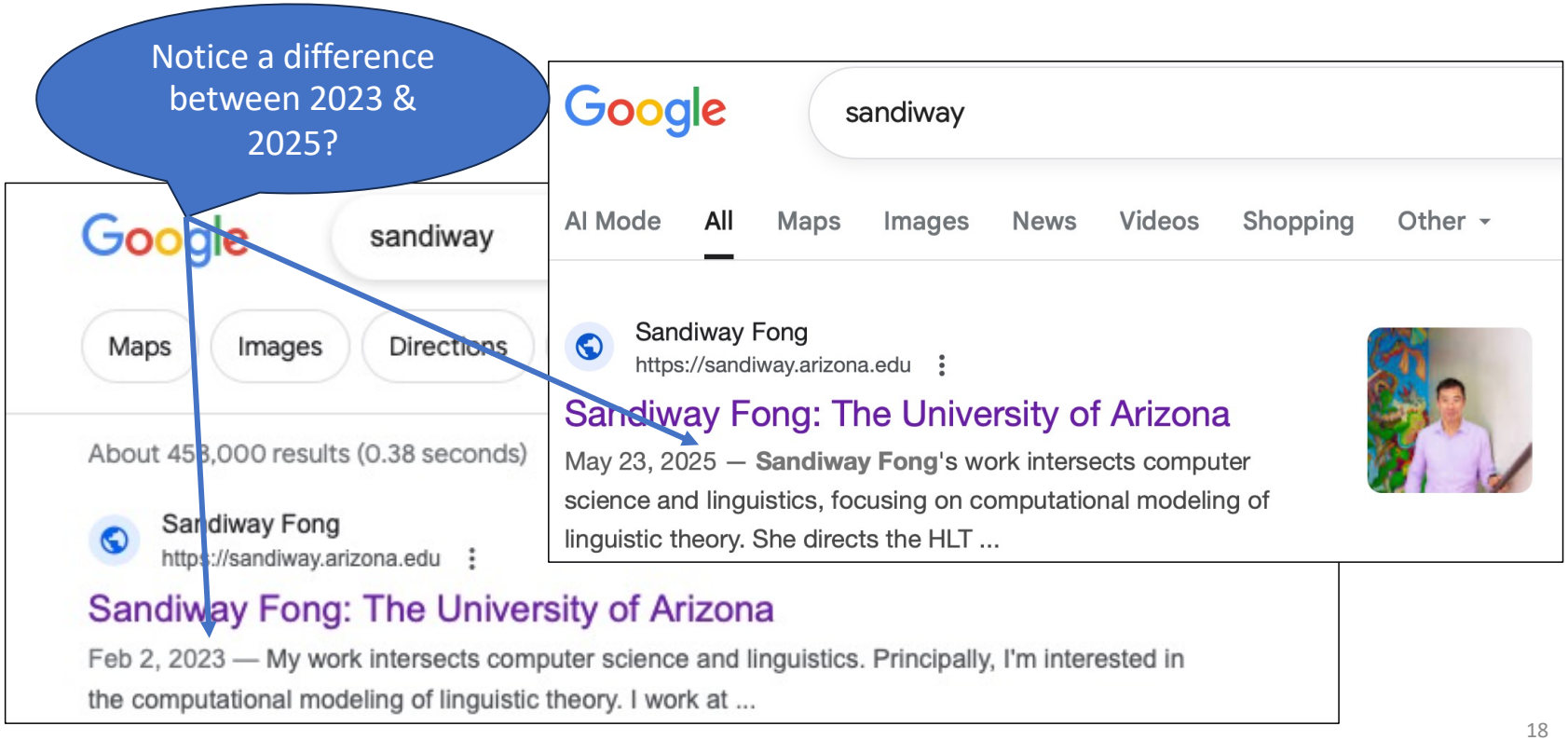
- Information contained in the course syllabus, other than the grade and absence policy, may be subject to change with advance notice, as deemed appropriate by the instructor.

- **Questions?**

UofA website

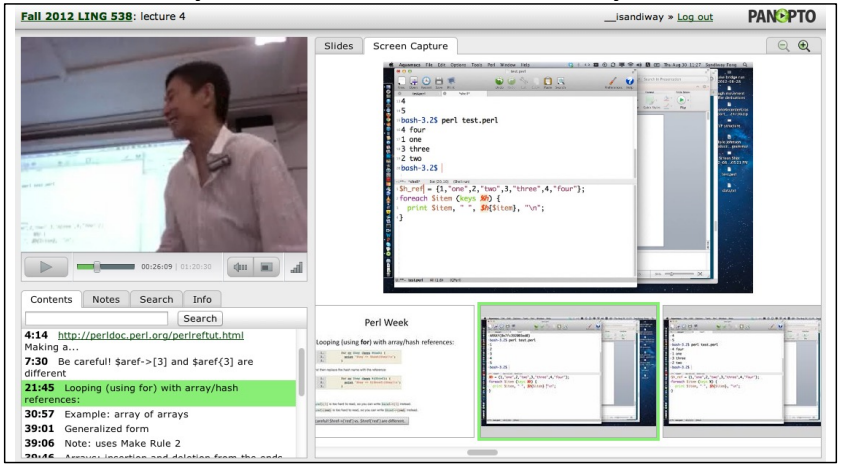
- Download lecture slides from my homepage
  - sandiway.arizona.edu
  - sandiway.arizona.edu/ling538-25
  - available from just before class time
    - (afterwards, please look again for updates and corrections)
  - in .pptx (good for animations) and .pdf formats

UofA website



Panopto

- Lectures will be recorded using the Panopto system
  - accessible via the course webpage and your browser
  - **sometimes crashes**
  - (video, laptop screen, synchronized slides, keyword search)

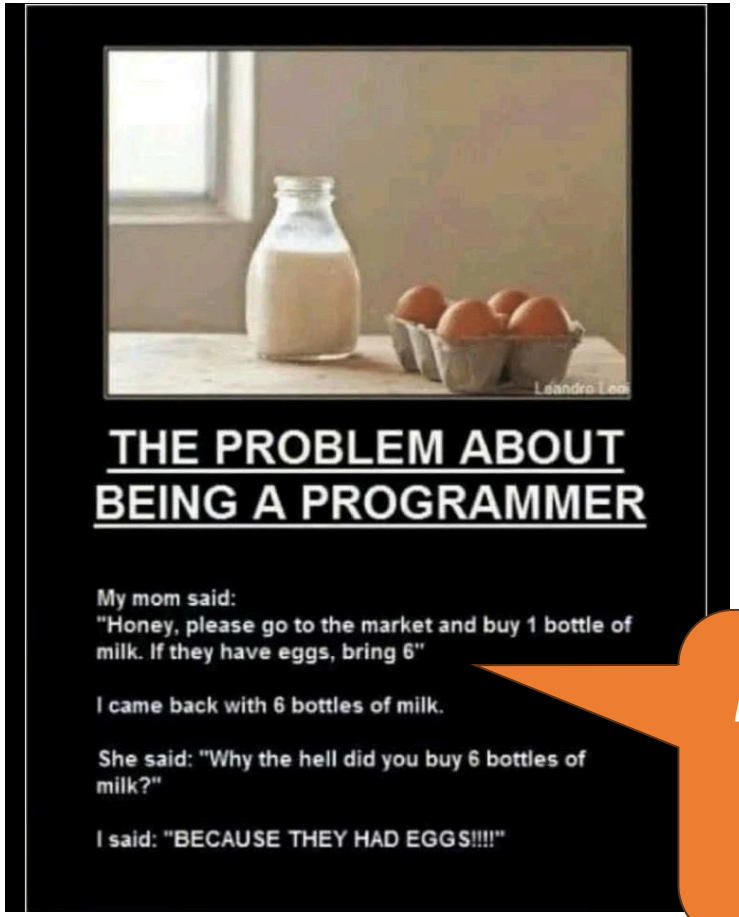


Syllabus

- I’m gonna assume you don’t know how to program at all (*yet*)
  - we’re going to use Perl and Python
  - good to learn both …
  - good to be *polyvalent*, you'll also get some Prolog

Being Precise

- Always check to see if you typed something in exactly right.
- *saw this on my Facebook feed*
- Do you understand the joke?



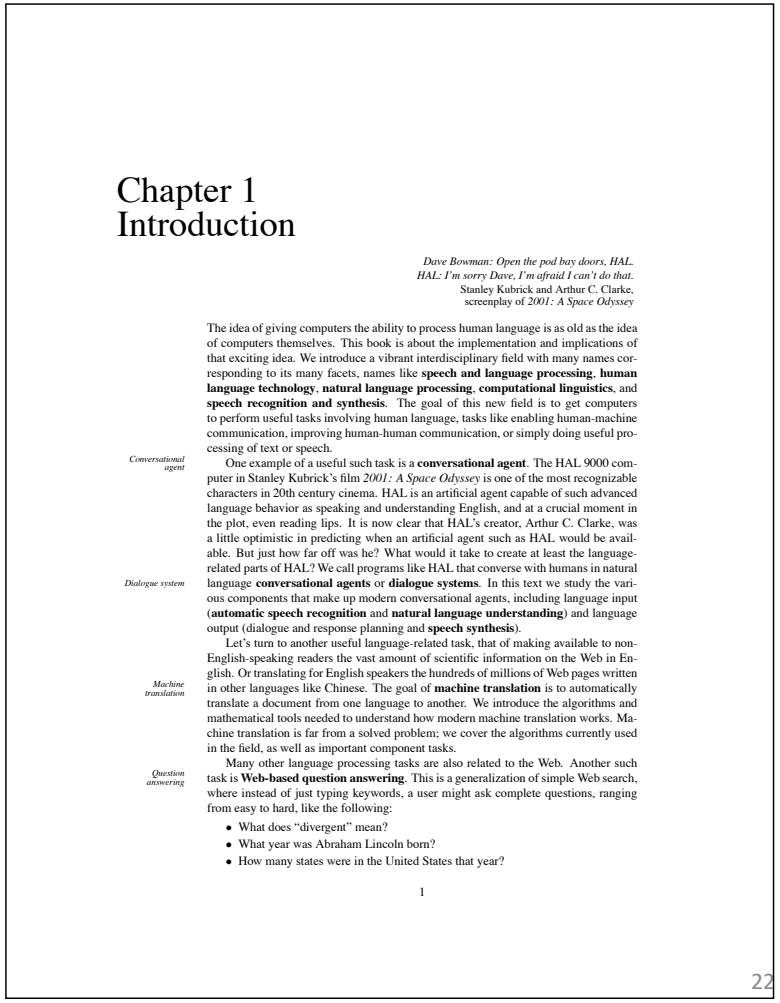
if condition: do command

Homework: Reading

Homework 1:

- Chapter 1 from JM (2nd ed)
  - **READ IT before next time!**
  - **in-class Quiz on Thursday**
  - available on course webpage as:

sandiway.arizona.edu/ling538-25/1.pdf



Chapter 1
Introduction

Dave Bowman: Open the pod bay doors, HAL.
HAL: I'm sorry Dave, I'm afraid I can't do that.
Stanley Kubrick and Arthur C. Clarke,
screenplay of *2001: A Space Odyssey*

The idea of giving computers the ability to process human language is as old as the idea of computers themselves. This book is about the implementation and implications of that exciting idea. We introduce a vibrant interdisciplinary field with many names corresponding to its many facets, names like **speech and language processing**, **human language technology**, **natural language processing**, **computational linguistics**, and **speech recognition and synthesis**. The goal of this new field is to get computers to perform useful tasks involving human language, tasks like enabling human-machine communication, improving human-human communication, or simply doing useful processing of text or speech.

Conversational agent

One example of a useful such task is a **conversational agent**. The HAL 9000 computer in Stanley Kubrick's film *2001: A Space Odyssey* is one of the most recognizable characters in 20th century cinema. HAL is an artificial agent capable of such advanced language behavior as speaking and understanding English, and at a crucial moment in the plot, even reading lips. It is now clear that HAL's creator, Arthur C. Clarke, was a little optimistic in predicting when an artificial agent such as HAL would be available. But just how far off was he? What would it take to create at least the language-related parts of HAL? We call programs like HAL that converse with humans in natural language **conversational agents** or **dialogue systems**. In this text we study the various components that make up modern conversational agents, including language input (**automatic speech recognition** and **natural language understanding**) and language output (dialogue and response planning and **speech synthesis**).

Dialogue system

Machine translation

Let's turn to another useful language-related task, that of making available to non-English-speaking readers the vast amount of scientific information on the Web in English. Or translating for English speakers the hundreds of millions of Web pages written in other languages like Chinese. The goal of **machine translation** is to automatically translate a document from one language to another. We introduce the algorithms and mathematical tools needed to understand how modern machine translation works. Machine translation is far from a solved problem; we cover the algorithms currently used in the field, as well as important component tasks.

Question answering

Many other language processing tasks are also related to the Web. Another such task is **Web-based question answering**. This is a generalization of simple Web search, where instead of just typing keywords, a user might ask complete questions, ranging from easy to hard, like the following:

- What does "divergent" mean?
- What year was Abraham Lincoln born?
- How many states were in the United States that year?

1

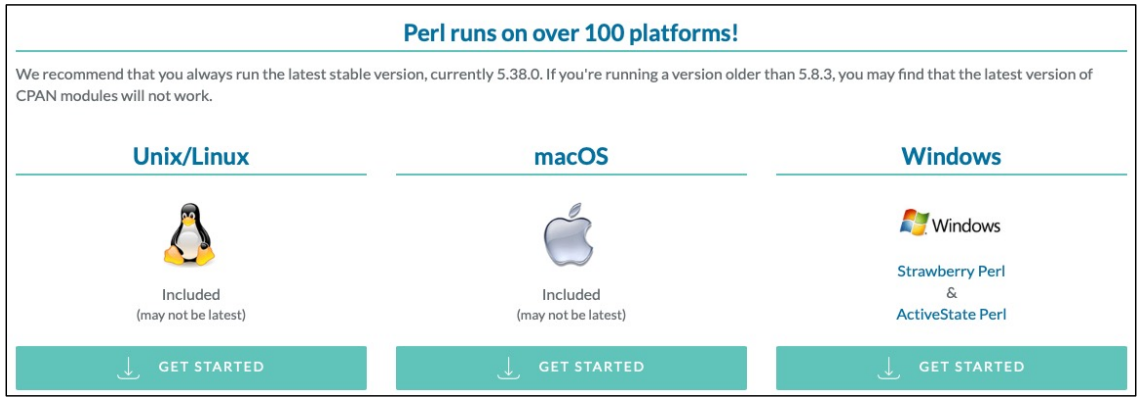
Homework 2

Could be the *null* homework for many of you:

- Install Perl, and
- Install Python (version 3.X, **not obsolete** 2.7)

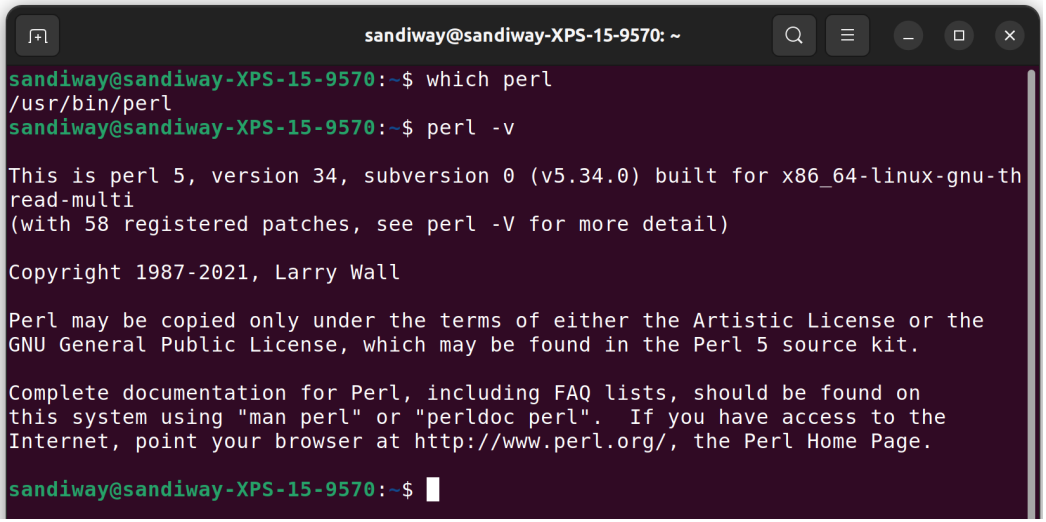
Homework 2: Install Perl

- Install Perl on your computer
  - pre-installed on macOS and Linux, check your machine from the Terminal/command line
  - on Windows PCs, if you don't already have it, it's freely available here
  - https://www.perl.org/get.html



Homework 2: Install Perl

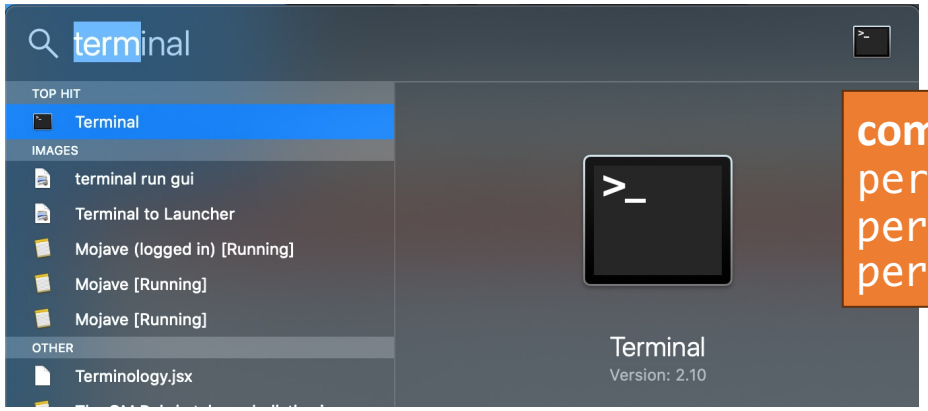
- How to check?
  - `which perl`
  - `perl –v`
- Ubuntu (Terminal):



Homework 2: Install Perl

- macOS (Terminal): (*complete path specified here*)

PATH
  - `/usr/bin/perl`
  - `/opt/homebrew/bin/perl` (Homebrew package manager *brew.sh*)
  - `/opt/local/bin/perl` (MacPorts package manager)



commands:

```
perl          (hangs; type Control-D)
perl -v
perl -V       (much more verbose)
```

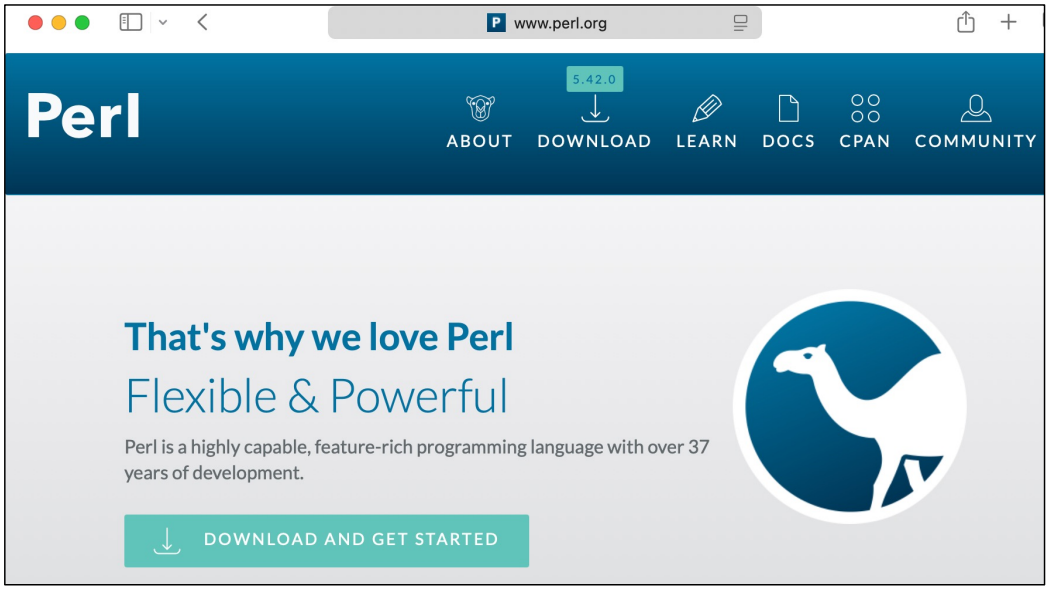
Homework 2: Install Perl

https://www.perl.org/learn.html

start here soon

Get Started

- learn.perl.org
- A brief introduction
- Free online Perl books
- Join your local community
- Books and More



Learning Perl

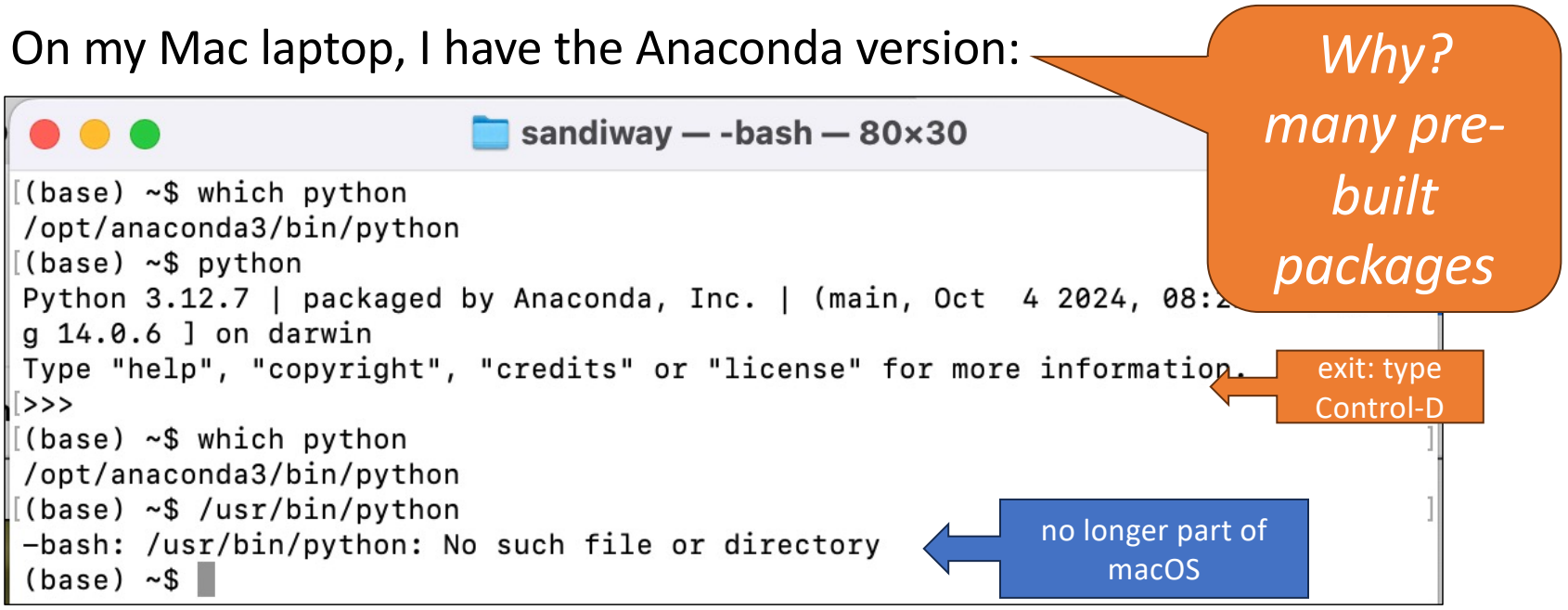
- Learn Perl
  - https://perldoc.perl.org/perlintro.html

Homework 2: Install Python

- www.python.org
- **Note**: 3.x is not backwards compatible with Python 2.7!

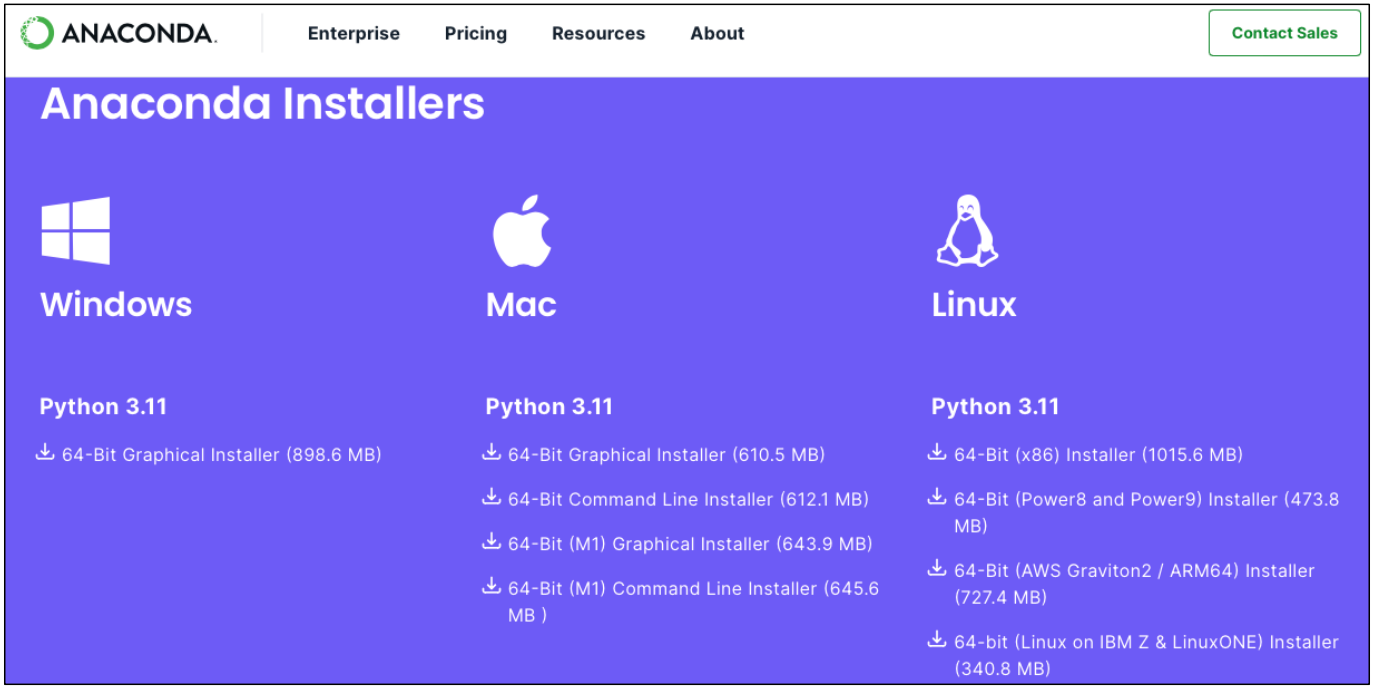
Homework 2: Install Python

- On my Mac laptop, I have the Anaconda version:



Homework 2: Install Python

https://www.anaconda.com/download

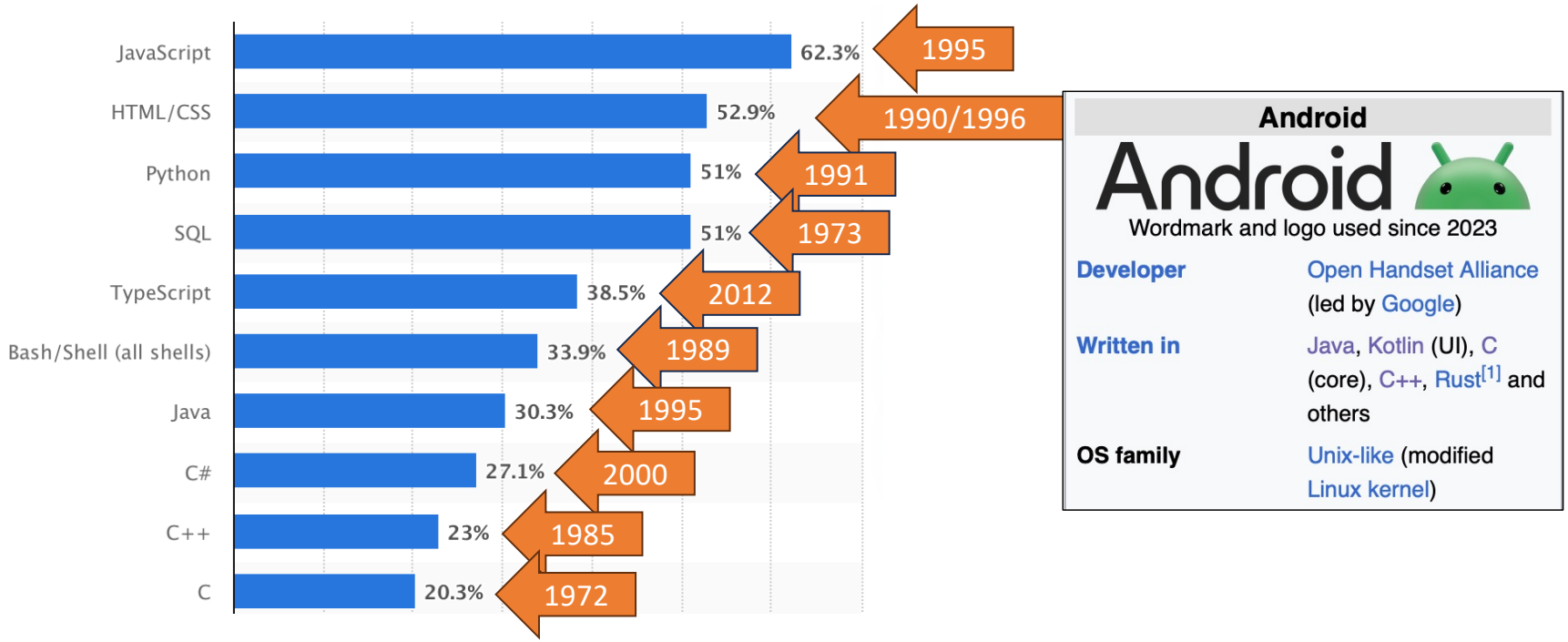


Which language is easier?

A subjective question …

- All good programmers know more than one programming language, always an advantage to be versatile.
- In NLP, Python is overwhelmingly popular, but we will do both Perl and Python. AND for writing grammars, Prolog.

Development work (2024 survey)



www.statista.com/statistics/793628/worldwide-developer-survey-most-used-languages/

How many programmers, how many lines?

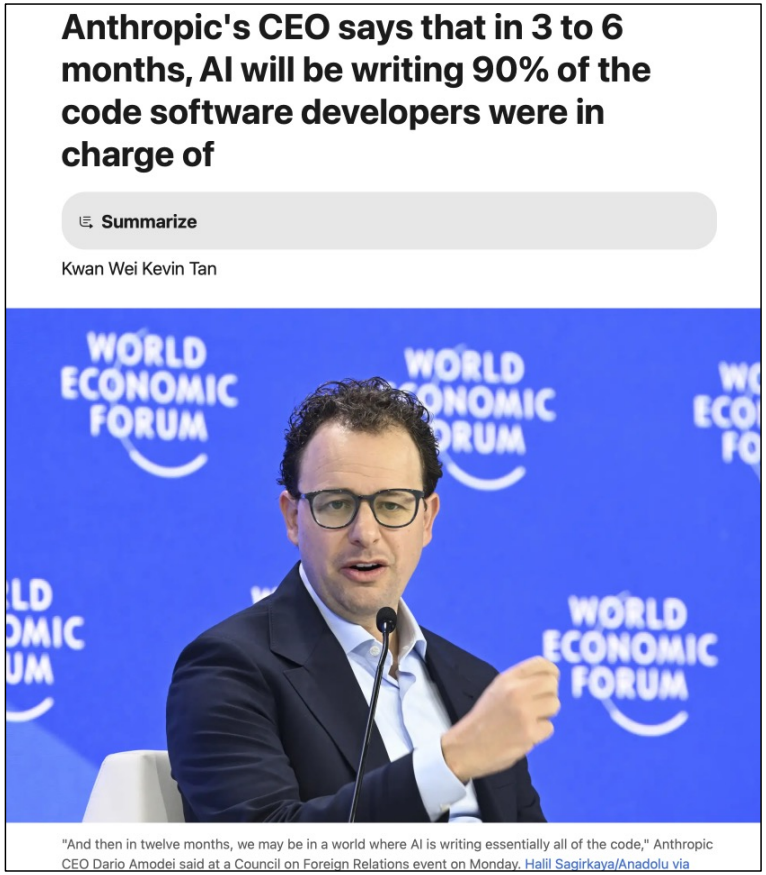
- medium.com/modern-stack/how-much-computer-code-has-been-written-c8c03100f459

> Back to the question of how many people are out there writing code. We have arrived at the educated guess that there are about 6 million people in the world writing code right now. The real number is probably much higher than this, but this number works for our calculation.

> **2,781,000,000,000**
>
> Roughly 2.8 Trillion Lines of Code have been written in the past 20 years.
>
> That is more than 5X the estimated number of stars in the Milky Way!

AI and Coding

www.businessinsider.com/anthropic-ceo-ai-90-percent-code-3-to-6-months-2025-3



AI and Coding

www.pcmag.com/news/ai-now-writes-over-25-percent-of-code-at-google